Improved Regularisation for Automatic Data Augmentation

CS 748 Project Proposal

About the Project

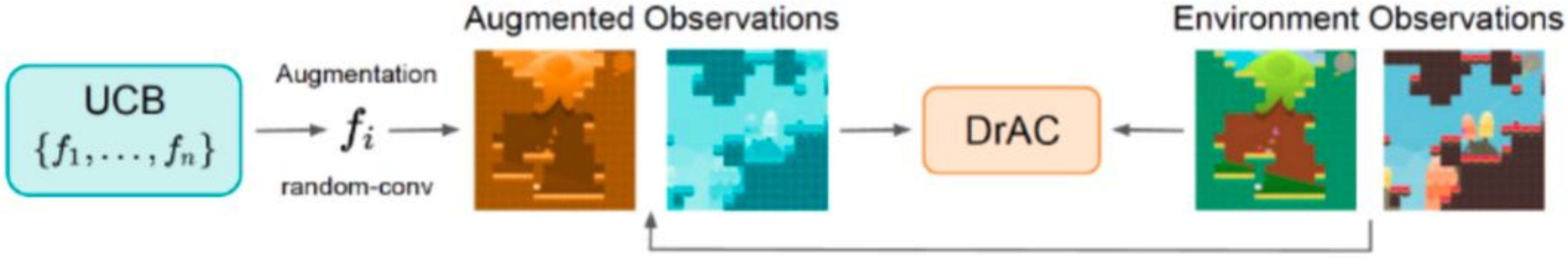
- Automatic Data Augmentation has been proposed to improve the performance of models by the means of better generalisation capabilities.
- **RAD**, followed by **DrAC**, both set state-of-art scores on the *ProcGen Benchmarks*
- We propose Improved Regularization for Automatic Data Augmentation for both the value as well as the policy function

ProcGen Environments

- 16 different games (environments)
- Measures both efficiency and generalisability of the model
- Each environment can generate distinct training and testing sets

Automatic Data Augmentation

- **DrAC** is an algorithm, given an image transformation (data augmentation), optimizes the policy and value function with regularization terms
- The paper proposes three methods to automate the process of finding the best data augmentations for training the algorithm
- We follow one of them: **UCB-DrAC**



DrAC

The base algorithm is an actor critic based PPO algorithm to learn an optimal policy and a value function given an MDP

DrAC proposes two regularization terms for the policy and value function: G_π and G_V respectively

UCB-DrAC

Algorithm 2 UCB-DrAC

Hyperparameters: Set of image transformations $\mathcal{F} = \{f^1, \ldots, f^n\}$, exploration coefficient c, window for estimating the Q-functions W, number of updates K, initial policy parameters π_θ, initial value function V_ϕ.

$N(f) = 1, \ \forall\, f \in \mathcal{F}$ ▷ Initialize the number of times each augmentation was selected

$Q(f) = 0, \ \forall\, f \in \mathcal{F}$ ▷ Initialize the Q-functions for all augmentations

$R(f) = \text{FIFO}(W), \ \forall\, f \in \mathcal{F}$ ▷ Initialize the lists of returns for all augmentations

for $k = 1, \ldots, K$ **do**

    $f_k = \text{argmax}_{f \in \mathcal{F}} \left[Q(f) + c \sqrt{\frac{\log(k)}{N(f)}} \right]$ ▷ Use UCB to select an augmentation

    Update the policy and value function according to Algorithm 1 with $f = f_k$ and $K = 1$:

    $\theta \leftarrow \arg\max_\theta J_{\text{DrAC}}$ ▷ Update the policy

    $\phi \leftarrow \arg\max_\phi J_{\text{DrAC}}$ ▷ Update the value function

    Compute the mean return obtained by the new policy r_k.

    Add r_k to the $R(f_k)$ list using the first-in-first-out rule.

    $Q(f_k) \leftarrow \frac{1}{|R(f_k)|} \sum_{r \in R(f_k)} r$

    $N(f_k) \leftarrow N(f_k) + 1$

end for

Project Goals

Goal 1

Replacing KL Divergence with JS Divergence

Goal 2

Replacing L2 Norm With Elastic Net Based Regularisation

Goal 3

Proposing Thompson Sampling for Selecting Augmentations

1: Replacing KL Divergence with JS Divergence

- The *Jensen-Shannon (JS) Divergence* is given as :

$$JS(P||Q) = 1/2 \times KL(P||M) + 1/2 \times KL(Q||M)$$ where $$M = (P+Q)/2$$

- UCB-DrAC uses KL Divergence for regularising the policy **G_π regularization term**.
- *KL(P || Q)* is known to punish any sample x that obeys *Q(x)=0* and *P(x)≠0*.
- The *JS Divergence* on the other hand is symmetric and much smoother than the *KL Divergence*.

2: Replacing L2 Norm With Elastic Net Based Regularisation

- L2 norm is known to punish the outliers heavily as compared to L1 norm.
- L1 norm is known to induce sparsity in the corresponding the network, which in turn encourages better generalisations in the model.
- We would also like to penalise any state value that is very far from the un-transformed state values to not affect the subsequent policy significantly.
- Hence we propose to use the elastic net regularisation instead of the current L2 regularisation in the **G_V regularisation term**.

3: Propose Thompson Sampling for Selecting Augmentations

- We would also like to explore more recent methods for selecting an augmentation at each step of training.
- Currently this is being done by using an epsilon-greedy strategy in DrAC algorithm.
- We would like to explore the use of Thompson Sampling which is known to achieve optimal regret (sub-linear) as compared to epsilon greedy which achieves linear regret.

Our Team

Aditya Badola - 190050006

Advait Kumar - 18D070003

Hitul Desai - 18D070009